[0045] What is claimed is:

1. A method for digital signature of an electronic file, the method comprising the steps of:

a) determining a portion of the electronic file that is to used for computing a digital signature; and

b) digitally signing a block of data that consists of the determined portion and creating the digital signature of the electronic file;

wherein the portion of the electronic file that is to be used for the digital signature is computed using one or more functions that are known to a signer of the electronic file who executes the digital signature.

2. The method claimed in claim 1, further comprising the step of:

c) prior to step b), computing a Message Digest (MD) value using the determined portion of the electronic file;

wherein the block of data that is digitally signed in step b) consists of the MD value.

3. The method claimed in claim 2, wherein step a) comprises the steps of:

a.1) dividing the electronic file into a plurality of blocks; and

a.2) from each block of the plurality of blocks, extracting a block portion and copying the block portion into a buffer;

wherein when the block portion is extracted and copied into the buffer for each block of the plurality of blocks, the buffer comprises the portion of the electronic file that is to be digitally signed.

4. The method claimed in claim 3, wherein step a.2) comprises the steps of:

a.2.1) for each block of the plurality of blocks,

computing a value *m* using the one or more functions; and

within a block of the plurality of blocks, at a location *m* bytes apart from a beginning of the block, extracting the block portion and copying the block portion into the buffer, wherein the block portion has *p* bytes of length.

5. The method claimed in claim 4, wherein for computing the value of *m*, a first function is applied on a shared secret key of the signer of the electronic file, and wherein a second function is applied on a result of the first function and on a variable *j* that represents the number of a current block from the plurality of blocks of the electronic file.

6. The method claimed in claim 2, further comprising the step of:

d) appending digital signature to the electronic file and creating a digitally signed electronic file.

7. The method claimed in claim 2, wherein the electronic file is a binary file.

8. The method claimed in claim 2, wherein the electronic file is an executable file.

9. The method claimed in claim 2, wherein the electronic file is a shared library file.

10. A method for digital signature verification of an electronic file, the method comprising the steps of:

a) extracting the digital signature from the electronic file;

b) determining a portion of the electronic file that was used for computing the digital signature;

c) decrypting the digital signature using a public key of the signer of the electronic file, and obtaining a block of data; and

d) comparing the portion of the electronic file that was used for computing the digital signature with the block of data for determining an authenticity and an integrity of the electronic file;

wherein the portion of the electronic file that was used for computing the digital signature is computed using one or more functions that are known to a verifier of the digital signature verification of the electronic file.

11. The method claimed in claim 10, further comprising the steps of:

e) subsequent to step b), computing a Message Digest (MD2) value using the determined portion of the electronic file;

wherein the block of data obtained in step c) comprises an MD1 value and wherein step d) comprises the step of:

d.1) comparing the MD1 value with the MD2 value.

12. The method claimed in claim 11, wherein step b) comprises the steps of:

b.1) dividing the electronic file into a plurality of blocks; and

b.2) from each block of the plurality of blocks, extracting a block portion and copying the block portion into a buffer;

wherein when the block portion is extracted and copied into the buffer for each block of the plurality of blocks, the buffer comprises the portion of the electronic file that was used for computing the digital signature.

13. The method claimed in claim 12, wherein step b.2) comprises the steps of:

b.2.1) for each block of the plurality of blocks,

computing a value *m* using the one or more functions; and

within a block of the plurality of blocks, at a location *m* bytes apart from a beginning of the block, extracting the block portion and copying the block portion into the buffer, wherein the block portion has *p* bytes of length.

14. The method claimed in claim 13, wherein for computing the value of *m*, a first function is applied on a shared secret key of the signer of the electronic file, and wherein a second function is applied on a result of the first function and on a variable *j* that represents the number of a current block from the plurality of blocks of the electronic file.

15. The method claimed in claim 11, wherein the electronic file is a binary file.

16. The method claimed in claim 11, wherein the electronic file is an executable file.

17. The method claimed in claim 11, wherein the electronic file is a shared library file.

18. The method claimed in claim 11, wherein if the MD1 value is equal to the MD2 value, it is concluded that the digital signature is valid and the electronic file is authentic and unmodified with respect to the electronic file that was digitally signed.

19. The method claimed in claim 11, wherein if MD1 value is not equal to MD2 value, it is concluded that the digital signature is invalid and that the electronic file is corrupted.

20. A computer-system operated software application for digitally signing an electronic file, the computer-system operated software application comprising:

a File Analyzer module determining a portion of the electronic file that is to be used for computing a digital signature; and

a Digital Signature Processing module digitally signing a block of data comprising the determined portion of the electronic file and creating a digital signature for the electronic file;

wherein the portion of the electronic file that is to be used for computing the digital signature is computed by the File Analyzer module using one or more functions that are known to a signer of the electronic file who executes the digital signature.

21. The computer-system operated software application further comprising:

a Message Digest module computing a Message Digest (MD) value using the determined portion of the electronic file, wherein the block of data that is digitally signed consists of the MD value.

22. The computer-system operated software application claimed in claim 21, further comprising:

a buffer connected to the File Analyzer;

wherein the File Analyzer acts to divide the electronic file into a plurality of blocks, and from each block of the plurality of blocks, extracts a block portion and copies the block portion into the buffer, wherein when a block portion is extracted and copied into the buffer from each block of the plurality of blocks, the buffer comprises the portion of the electronic file that is to be digitally signed.

23. The computer-system operated software application claimed in claim 22, wherein for each block of the plurality of blocks, the File Analyzer module computes a value *m* using the one or more functions, and within a block of the plurality of blocks, at a location *m* bytes apart from a beginning of the block, extracts the block portion and copies the block portion into the buffer.

24. The computer-system operated software application claimed in claim 23, wherein for computing the value of *m*, the File Analyzer module applies a first function on a shared secret key of the signer of the electronic file, and further applies a second function on a result of the first function and on a variable *j* that represents the number of a current block from the plurality of blocks of the electronic file.

25. The computer-system operated software application claimed in claim 21, wherein the File Analyzer module appends the signed MD value to the electronic file and creates a digitally signed electronic file.

26. The computer-system operated software application claimed in claim 21, wherein the electronic file is a binary file.

27. The computer-system operated software application claimed in claim 21, wherein the electronic file is an executable file.

28. The computer-system operated software application claimed in claim 21, wherein the electronic file is a shared library file.

29. A computer-system operated software application for digital signature verification of an electronic file, comprising:

a File Analyzer module extracting a digital signature from the electronic file, and determining a portion of the electronic file that was used for computing the digital signature; and

a Digital Signature Processing module decrypting the digital signature using a public key of the signer of the electronic file, and obtaining a block of data that was used for computing the digital signature;

wherein the Digital Signature Processing Module compares the portion of the electronic file that was used for computing the digital signature with the block of data for determining an authenticity and an integrity of the electronic file, wherein the portion of the electronic file that was used for computing the digital signature is computed using one or more functions that are known to a verifier of the digital signature verification of the electronic file.

30. The computer-system operated software application claimed in claim 29, further comprising:

a Message Digest module computing a Message Digest (MD2) value using the determined portion of the electronic file and sending the MD2 value to the Digital Signature Processing module;

wherein the block of data comprises an MD1 value and wherein the Digital Signature Processing module acts to compare the MD1 value with the MD2 value for determining an authenticity and an integrity of the electronic file.

31. The computer-system operated software application claimed in claim 30, wherein the File Analyzer divides the electronic file into a plurality of blocks, and from each block of the plurality of blocks, extracts a block portion and copies the block portion into a buffer, wherein when the block portion is extracted and copied into the buffer for each block of the plurality of blocks, the buffer comprises the portion of the electronic file that was used for the digital signature.

32. The computer system operated software application claimed in claim 31, wherein for each block of the plurality of blocks, the File Analyzer module computes a value *m* using the one or more functions, and within a block of the plurality of blocks, at a location *m* bytes apart from a beginning of the block, extracts the block portion and copies the block portion into the buffer.

33. The computer system operated software application claimed in claim 32, wherein for computing the value of *m*, the File Analyzer applies a first function on a shared secret key of the signer of the electronic file, and further applies a second function on a result of the first function and on a variable *j* that represents the number of a current block from the plurality of blocks of the electronic file.

34. The computer-system operated software application claimed in claim 30, wherein the electronic file is a binary file.

35. The computer-system operated software application claimed in claim 30, wherein the electronic file is an executable file.

36. The computer-system operated software application claimed in claim 30, wherein the electronic file is a shared library file.

37. The computer-system operated software application claimed in claim 30, wherein the Digital Signature Processing module concludes that the digital signature is valid and the electronic file is authentic and unmodified with respect to the electronic file that was digitally signed if the MD1 value is equal to the MD2 value.

38. The computer-system operated software application claimed in claim 30, wherein the Digital Signature Processing module concludes that the digital signature is invalid and that the electronic file is corrupted if the MD1 value is not equal to the MD2 value.